In [1]:

```
# importing various libraries which are used
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import preprocessing as p
from scipy.stats import f
from sklearn.metrics import mean_squared_error, mean_absolute_error ,r2_score
```

In [2]:

```
# taking the csv file as input to create the model
file=input("csv file_name : ")
```

```
csv file_name : 525.csv
```

In [3]:

```
# creating a pandas dataframe using the values obtained in the csv
df=pd.read_csv(file)
```

In [4]:

```
# printing the head of the data frame to get the gist of the values
print("\n data frame head :- \n",df.head())
```

```
 data frame head :-
        time  cpu-cycles  instructions  l1d.replacement  icache_64b.if
tag_miss  \
0  0.100147   456755592    1291362820          7513808
570441
1  0.200376   459480437    1832441076           345008
52418
2  0.300557   459304880    1838202506           562170
47255
3  0.400721   459148806    1827730167           387356
50644
4  0.500879   459179614    1851529988           499468
44499

   l2_rqsts.all_demand_miss  longest_lat_cache.miss  \
0                   2856236                 3472090
1                    119869                  118248
2                    122919                  135420
3                    128058                  117967
4                    159659                  152505

   br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                      49302590                         711
1                      50959979                           2
2                      51311981                          42
3                      50727322                          36
4                      55496246                           1

   itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                        2197                            14554
1                         103                             2102
2                         145                             2314
3                         147                             2309
4                         119                             3754

   dtlb_store_misses.walk_completed  branch-misses
0                            466779          56042
1                               631          30032
2                               710          51011
3                               706          72160
4                               884          78504
```

In [5]:

```
# creating a new col in our data frame which consists of the CPI per tuple
df[['CPI']]=df[['cpu-cycles']].div(df['instructions'], axis=0)
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100147   456755592    1291362820          7513808
1       0.200376   459480437    1832441076           345008
2       0.300557   459304880    1838202506           562170
3       0.400721   459148806    1827730167           387356
4       0.500879   459179614    1851529988           499468
...          ...         ...           ...              ...
1161  116.432215   459147283    1749656078          3854475
1162  116.532424   459207683    1721306411          3167034
1163  116.632633   459080351    1753418024          3753572
1164  116.732805   458911898    1733096903          3703142
1165  116.827435   433443417    1598797953          2991259

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                    570441                   2856236
3472090
1                     52418                    119869
118248
2                     47255                    122919
135420
3                     50644                    128058
117967
4                     44499                    159659
152505
...                     ...                       ...
...
1161                3043637                    167023
321945
1162                2815048                    218325
308488
1163                2843937                    217580
222150
1164                3185560                    167850
291261
1165                3560431                    543651
897469

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                         49302590                         711
1                         50959979                           2
2                         51311981                          42
3                         50727322                          36
4                         55496246                           1
...                            ...                         ...
1161                      58494619                           7
1162                      67755760                          27
1163                      67163451                         148
1164                      58374015                           3
1165                      68016219                        1805

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                           2197                            14554
1                            103                             2102
2                            145                             2314
3                            147                             2309
4                            119                             3754
...                          ...                              ...
1161                         120                             2802
1162                         135                             2832
1163                         427                             2342
```

```
1164                          199                                2497
1165                         6938                                5044

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                               466779          56042  0.353700
1                                  631          30032  0.250748
2                                  710          51011  0.249866
3                                  706          72160  0.251213
4                                  884          78504  0.248000
...                                ...            ...       ...
1161                              1318        1162079  0.262421
1162                              1372        1340702  0.266779
1163                              1571        1286765  0.261820
1164                              1399        1307049  0.264793
1165                              2186        1258590  0.271106

[1166 rows x 14 columns]
```

In [6]:

```
# dividing all the values by instruction so that we get values in each coloumn per i
df[['l1d.replacement','icache_64b.iftag_miss','l2_rqsts.all_demand_miss','longest_la
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100147   456755592    1291362820         0.005819
1       0.200376   459480437    1832441076         0.000188
2       0.300557   459304880    1838202506         0.000306
3       0.400721   459148806    1827730167         0.000212
4       0.500879   459179614    1851529988         0.000270
...          ...         ...           ...              ...
1161  116.432215   459147283    1749656078         0.002203
1162  116.532424   459207683    1721306411         0.001840
1163  116.632633   459080351    1753418024         0.002141
1164  116.732805   458911898    1733096903         0.002137
1165  116.827435   433443417    1598797953         0.001871

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                  0.000442                  0.002212
0.002689
1                  0.000029                  0.000065
0.000065
2                  0.000026                  0.000067
0.000074
3                  0.000028                  0.000070
0.000065
4                  0.000024                  0.000086
0.000082
...                     ...                       ...
...
1161               0.001740                  0.000095
0.000184
1162               0.001635                  0.000127
0.000179
1163               0.001622                  0.000124
0.000127
1164               0.001838                  0.000097
0.000168
1165               0.002227                  0.000340
0.000561

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                         0.038179                5.505811e-07
1                         0.027810                1.091440e-09
2                         0.027914                2.284841e-08
3                         0.027754                1.969656e-08
4                         0.029973                5.400939e-10
...                            ...                         ...
1161                      0.033432                4.000786e-09
1162                      0.039363                1.568576e-08
1163                      0.038304                8.440657e-08
1164                      0.033682                1.731005e-09
1165                      0.042542                1.128973e-06

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                  1.701303e-06                         0.000011
1                  5.620917e-08                         0.000001
2                  7.888141e-08                         0.000001
3                  8.042763e-08                         0.000001
4                  6.427117e-08                         0.000002
...                         ...                              ...
1161               6.858491e-08                         0.000002
1162               7.842880e-08                         0.000002
1163               2.435244e-07                         0.000001
```

```
1164                 1.148234e-07                          0.000001
1165                 4.339510e-06                          0.000003

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                           3.614623e-04       0.000043  0.353700
1                           3.443494e-07       0.000016  0.250748
2                           3.862469e-07       0.000028  0.249866
3                           3.862715e-07       0.000039  0.251213
4                           4.774430e-07       0.000042  0.248000
...                                  ...            ...       ...
1161                        7.532909e-07       0.000664  0.262421
1162                        7.970690e-07       0.000779  0.266779
1163                        8.959643e-07       0.000734  0.261820
1164                        8.072255e-07       0.000754  0.264793
1165                        1.367277e-06       0.000787  0.271106

[1166 rows x 14 columns]
```

In [7]:

```
# droping values such as time , instructions , cpu-cycles and br_inst_retired.all_br
df= df.drop(['time'], axis=1)
df= df.drop(['instructions'], axis=1)
df= df.drop(['cpu-cycles'], axis=1)
df= df.drop(['br_inst_retired.all_branches'], axis=1)
```

In [8]:

```
# assigning y as the CPI and then droping it from the dataframe
y=df['CPI']
df= df.drop(['CPI'], axis=1)
print("y values :- \n",y)
```

```
y values :-
 0       0.353700
1       0.250748
2       0.249866
3       0.251213
4       0.248000
          ...
1161    0.262421
1162    0.266779
1163    0.261820
1164    0.264793
1165    0.271106
Name: CPI, Length: 1166, dtype: float64
```

In [9]:

```
# assigning x as the dataframe
x=df
print("x values :- \n",x)
```

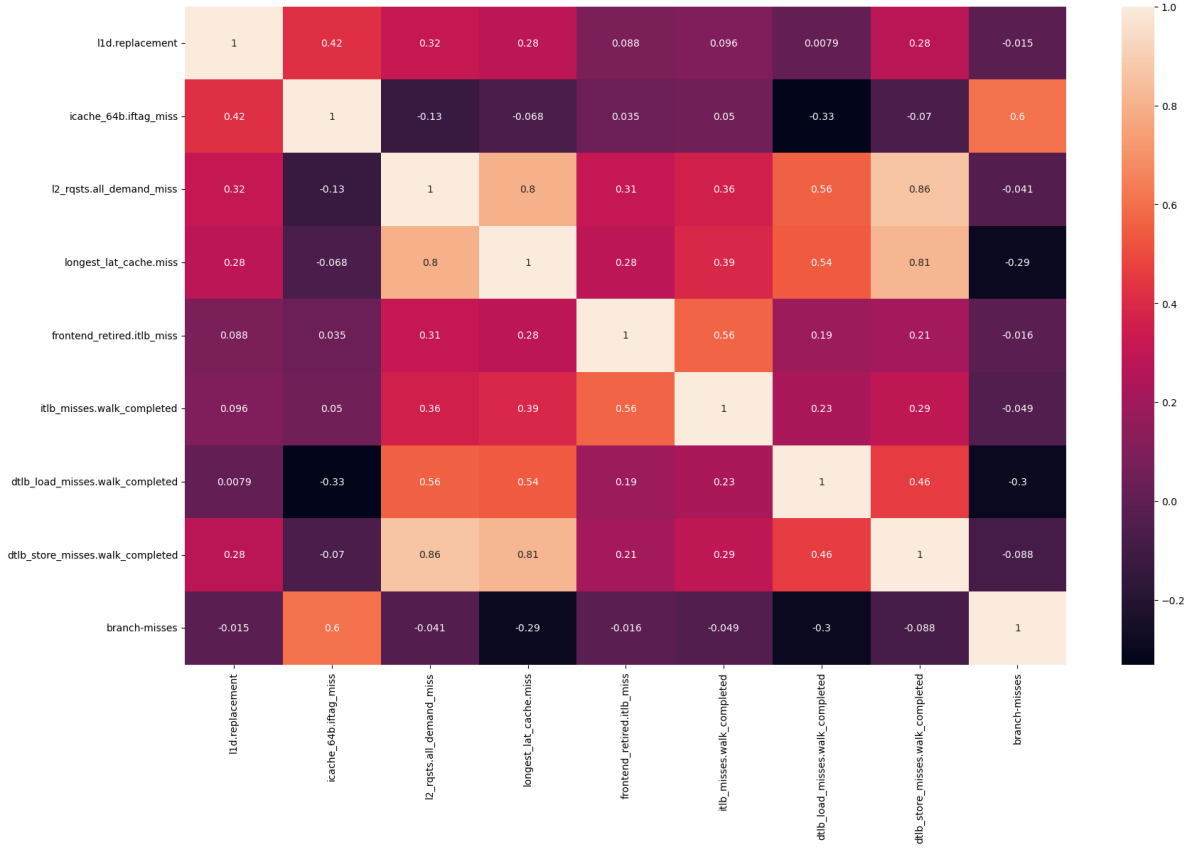
```
x values :-
       l1d.replacement  icache_64b.iftag_miss  l2_rqsts.all_demand_mis
s  \
0             0.005819               0.000442                  0.002212
1             0.000188               0.000029                  0.000065
2             0.000306               0.000026                  0.000067
3             0.000212               0.000028                  0.000070
4             0.000270               0.000024                  0.000086
...                ...                    ...                       ...
1161          0.002203               0.001740                  0.000095
1162          0.001840               0.001635                  0.000127
1163          0.002141               0.001622                  0.000124
1164          0.002137               0.001838                  0.000097
1165          0.001871               0.002227                  0.000340

      longest_lat_cache.miss  frontend_retired.itlb_miss  \
0                   0.002689                5.505811e-07
1                   0.000065                1.091440e-09
2                   0.000074                2.284841e-08
3                   0.000065                1.969656e-08
4                   0.000082                5.400939e-10
...                      ...                         ...
1161                0.000184                4.000786e-09
1162                0.000179                1.568576e-08
1163                0.000127                8.440657e-08
1164                0.000168                1.731005e-09
1165                0.000561                1.128973e-06

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                   1.701303e-06                         0.000011
1                   5.620917e-08                         0.000001
2                   7.888141e-08                         0.000001
3                   8.042763e-08                         0.000001
4                   6.427117e-08                         0.000002
...                          ...                              ...
1161                6.858491e-08                         0.000002
1162                7.842880e-08                         0.000002
1163                2.435244e-07                         0.000001
1164                1.148234e-07                         0.000001
1165                4.339510e-06                         0.000003

      dtlb_store_misses.walk_completed  branch-misses
0                         3.614623e-04       0.000043
1                         3.443494e-07       0.000016
2                         3.862469e-07       0.000028
3                         3.862715e-07       0.000039
4                         4.774430e-07       0.000042
...                                ...            ...
1161                      7.532909e-07       0.000664
1162                      7.970690e-07       0.000779
1163                      8.959643e-07       0.000734
1164                      8.072255e-07       0.000754
1165                      1.367277e-06       0.000787

[1166 rows x 9 columns]
```

In [10]:

```
# creating a heatmap of the correlation matrix
fig,axis = plt.subplots(figsize = (20,12))
sns.heatmap(x.corr(),annot=True)
```

Out[10]:

<AxesSubplot:>



In [11]:

```
# dividing the data set into test and train set in a 20:80 ration with a random stat
# the model trains on a particular set of values on every execution
X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=.20,random_state=55)
```

In [12]:

```
# using MinMax Scaler to scale the data within the given range of 0 to 1 such that
#shape of the original distribution is same after transformation
mms = p.MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.transform(X_test)
```

In [13]:

```
print("X_train :-\n",X_train)
```

```
X_train :-
 [[0.36811395 0.40711056 0.05408414 ... 0.0374567  0.00052425 0.187684
89]
 [0.25865359 0.48904693 0.07868011 ... 0.17654995 0.00300133 0.2112293
5]
 [0.32427497 0.28000777 0.0843554  ... 0.1604694  0.00137449 0.2003299
2]
 ...
 [0.22467181 0.33582204 0.06794616 ... 0.12148272 0.00290045 0.1346703
6]
 [0.34701448 0.44797971 0.07836278 ... 0.04692843 0.00240959 0.2039227
5]
 [0.2415527  0.45348752 0.02978432 ... 0.02110772 0.00056948 0.2973956
1]]
```

In [14]:

```
# mean of all the columns of the training set
df2 = X_train.mean(axis=0)
print(df2)
```

```
[0.27637432 0.3967009  0.04370873 0.03844749 0.04352056 0.0200822
 0.08024605 0.00309252 0.19957674]
```

In [15]:

```
# creating a linear regression model using sklearn.linear_model
model = LinearRegression(positive=True)
model.fit(X_train,y_train)
```

Out[15]:

```
LinearRegression(positive=True)
```

In [16]:

```
# finding the coefficients given by our model
c=model.coef_
print("\nCoefficients :- \n",c)
```

```
Coefficients :-
 [0.         0.00396553 0.         0.01738586 0.00474658 0.
 0.02014103 0.0755838  0.13307467]
```

In [17]:

```
# model intercept i.e. the " Base CPI "
i=model.intercept_
print("\nBase CPI : ",i)
```

```
Base CPI :  0.23867128791636505
```

In [18]:

```
# making the predictions using our model on the test set
predictions = model.predict(X_test)
```

In [19]:

```
# Actual CPI
ACPI = y_test.mean()
print("\n Actual CPI : ",ACPI)
```

```
 Actual CPI :  0.26958410180572984
```

In [20]:

```
# Predicted CPI
PCPI = predictions.mean()
print("\n Predicted CPI : ",PCPI)
```

```
 Predicted CPI :  0.2697672368110619
```

In [21]:

```
# Finding out RMSE , R^2 , adjusted R^2 using our predictions and test set
MSE = mean_squared_error(y_test, predictions)
rint("\n RMSE : ",RMSE)

2 = r2_score(y_test, predictions)
rint("\n R^2 : ",r2_score(y_test, predictions))

djusted_r2 = 1 - ( 1-model.score(X_test,y_test) ) * ( len(y_test) - 1 ) / ( len(y_tes
rint("\n adjusted R^2 : ",adjusted_r2)
```

```
 RMSE :  1.0161230065284389e-05

 R^2 :  0.8363157388978956

 adjusted R^2 :  0.8297391391214717
```

In [22]:

```
# finding absolute error and accuracy on test set
err = mean_absolute_error(y_test, predictions)
print ( "\n Test error is :" , err *100 , "% " )
print ( "\n Test Accuracy is :" , (1- err) *100 , "%" )
```

```
 Test error is : 0.2517300064142276 %

 Test Accuracy is : 99.74826999358577 %
```

In [23]:

```
# F-statistic value which should be > 2.5 and p-value which should be < 0.05
F = (R2/(1-R2))*((X_test.shape[0]-1-X_test.shape[1])/X_test.shape[1])
print("\n F-statistic : ",F)

p = 1-f.cdf(F,X_test.shape[1],(X_test.shape[0]-1-X_test.shape[1]))
print("\n p-value : ",p)
```

```
 F-statistic :  127.16536923775821

 p-value :  1.1102230246251565e-16
```

In [24]:

```
#no of coefficients
X_test.shape[1]
```

Out[24]:

```
9
```

In [25]:

```
# no of tuples in the test set
X_test.shape[0]
```

Out[25]:

```
234
```

In [26]:

```
# finding the residual for our test set
residuals = y_test - predictions
print("\n Residual :- \n ",residuals)
```

```
 Residual :-
  1043    0.000528
207     0.001219
805     0.005031
181     0.000717
770     0.002410
          ...
367    -0.003827
506    -0.000301
925    -0.003097
1004   -0.000089
904    -0.002492
Name: CPI, Length: 234, dtype: float64
```

In [27]:

```
# residual graph
data = {
    'predicted': [i for i in predictions],
    'residuals': [i for i in residuals]
}

dfr = pd.DataFrame(data)
sns.scatterplot(data=dfr, x="predicted", y="residuals")
```

Out[27]:

```
<AxesSubplot:xlabel='predicted', ylabel='residuals'>
```